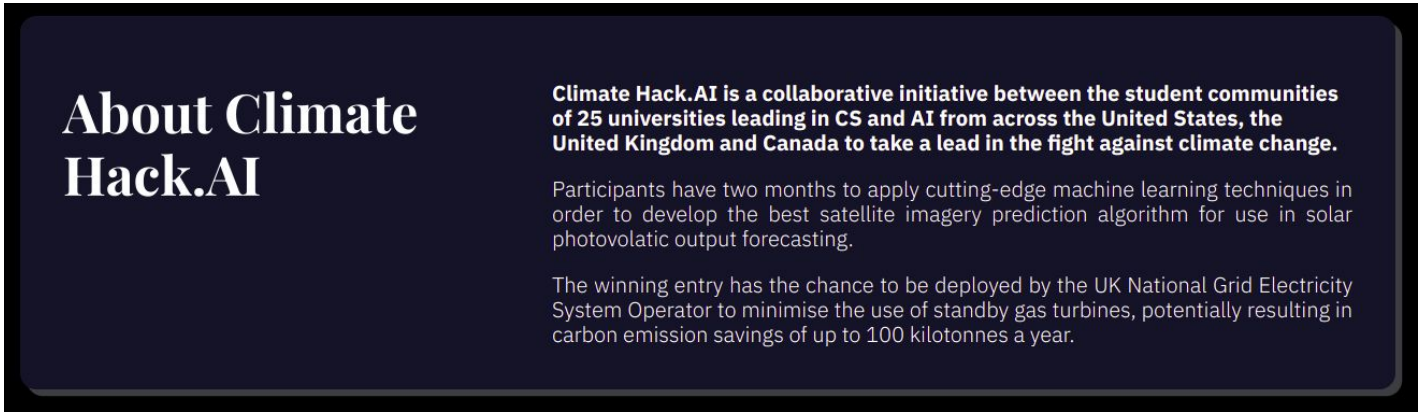
Logistic Regression

Intro to Machine Learning: Beginner Track #4

Feedback form: bit.ly/btrack-w22-feedback
Discord: bit.ly/ACMdiscord

acm acm.ai

Climate Hackathon



Intuition

Motivation

- Linear Regression: predict **continuous** data (eg: house price)

Output: $250,000

- Logistic Regression: Classify **categorical** data (this or that)

Output: Cat

An Example Problem

- Suppose we want to predict if it is going to rain today or not.
- What kind of features would we look at?
- Given these *features* how would we determine whether it is going to rain?
- **Logistic Regression!**

Example Problem

Suppose we are given the following data:

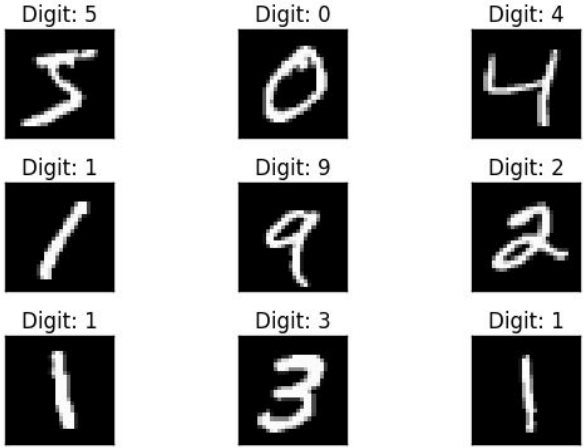
- Location: Los Angeles
- Temperature: 105F
- Cloud cover : Low
- Humidity: 50%
- Wind Speed: 10 mph

Is it more likely to rain or not to rain?

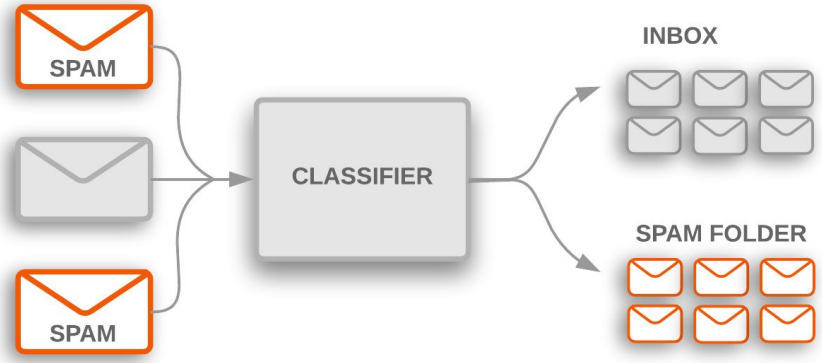
It's probably not going to rain!
i.e. the probability of it raining is less than **50%**

Examples of Classification Tasks

- Numerical digits
- MNIST dataset

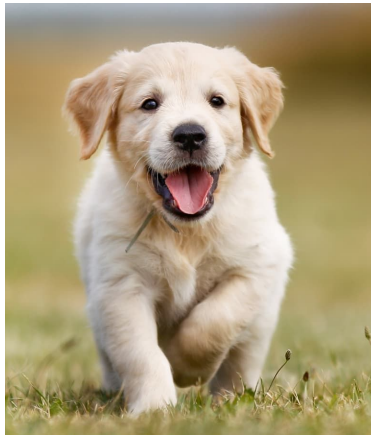


- Email Spam Detection
- Determine if an email is spam or normal email



Input and Output for Binary Classification

- Image Classification Input:
  - Array of pixels - the image
  - Each pixel is a feature



- Image Classification Output:
  - Binary Classification: 2 classes
  - Class label: **0** or **1**

Cool Binary Classification Demo!

https://tinyurl.com/btrack-w22-playground

This particular demo uses a neural network consisting of a single neuron which is actually the same a logistic regression model
Take a look at the visualization!

More on the Output

- Our final output is 0 or 1.
- To do so, we want some notion of “how sure we are” that the predicted class is actually 1.
    - i.e., we want the **probability** of the class being 1
- So our model will output the **probability of the class being 1**
    - Then we can simply say that if the probability < 0.5, we classify the object as class 0 and if the probability is > 0.5, we classify the object as class 1

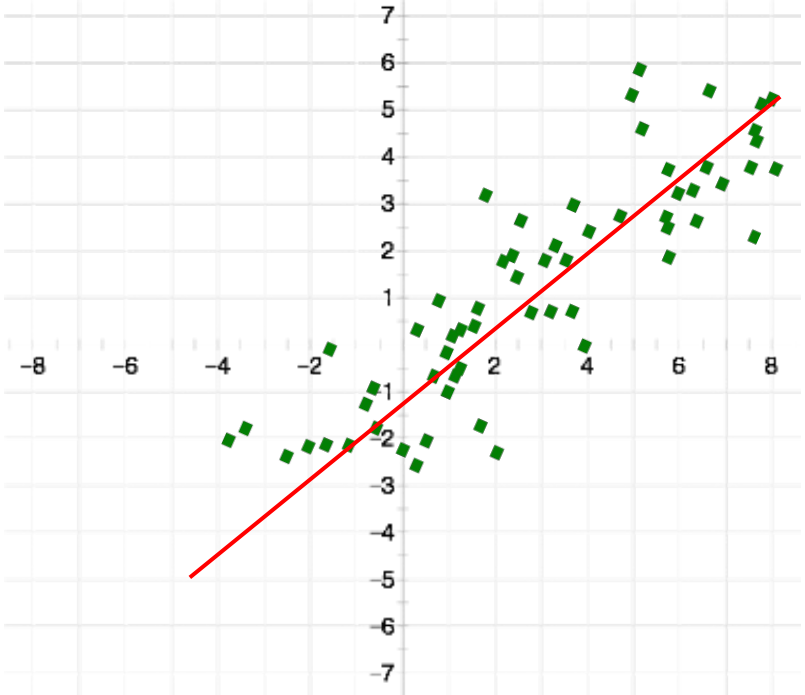
The Objective

- Given the **input features** of an object**,** find a **model** that predicts the **probability** that the object belongs to class **1**
  - Aside : How do we get the probability of object belonging to class 0 from this?
- We need a **hypothesis** to find this probability
  - What is a hypothesis?
- For these kinds of **classification problems,** one potential hypothesis is the **logistic regression model**

Building Off Linear Regression

Logistic Regression

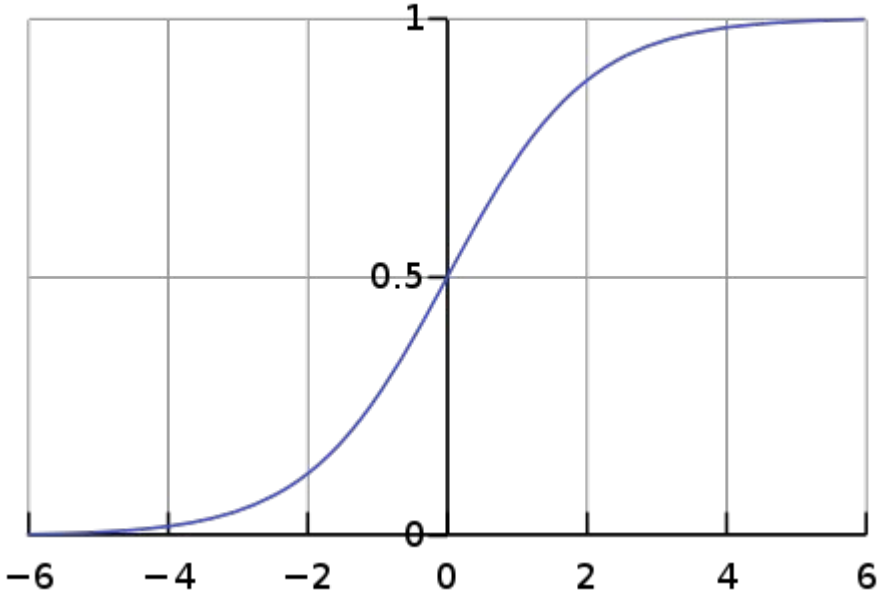
- We want our hypothesis to be a function to output a **probability (** b/w 0 and 1)
- Our **linear** function maps to any real number (-infinity, infinity)
- How do we fix this?



Solution: The Sigmoid Function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- If x is negative, σ(x) < 0.5
- If x is positive, σ(x) > 0.5
- Also, 0 < σ(x) < 1



Discussion Questions: Sigmoid Function

- Is this output continuous?
- How can we use this for classification?

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Building the Model

Wait, where are the weights?

- x is the input to the function
- But similar to linear regression, we need some **parameters** or weights to *optimize*
- How do we introduce these weights?

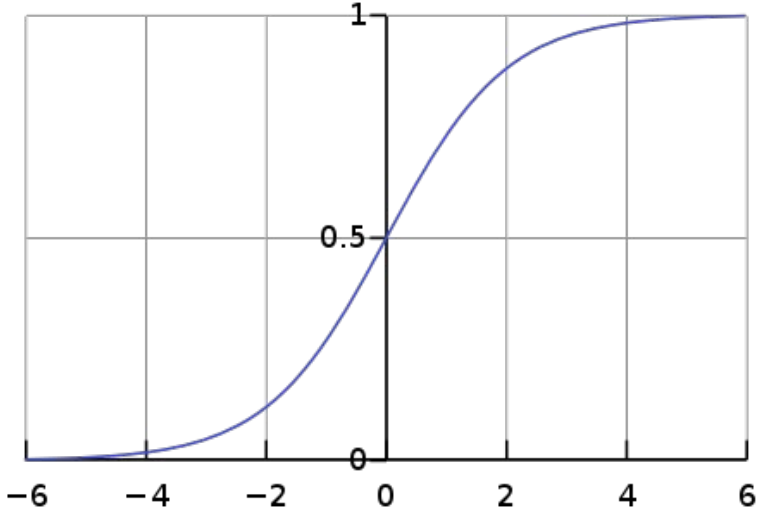
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

The Hypothesis

Instead of x, let's use our old linear function as the input

$$h(x) = X \cdot W + b$$

$$\hat{y}(x) = \sigma(X \cdot W + b) = \frac{1}{1 + e^{-(X \cdot W + b)}}$$



So depending on the values of **W** and **b**, an input **X** will result in a prediction ŷ that is either greater than 0.5 or lesser than 0.5

If ŷ<0.5 we can classify it as **0**

If ŷ>0.5 we can classify it as **1**

Probability of being a particular class

- Think of the output of the model as the **probability** of the input being class 1 given the features X.

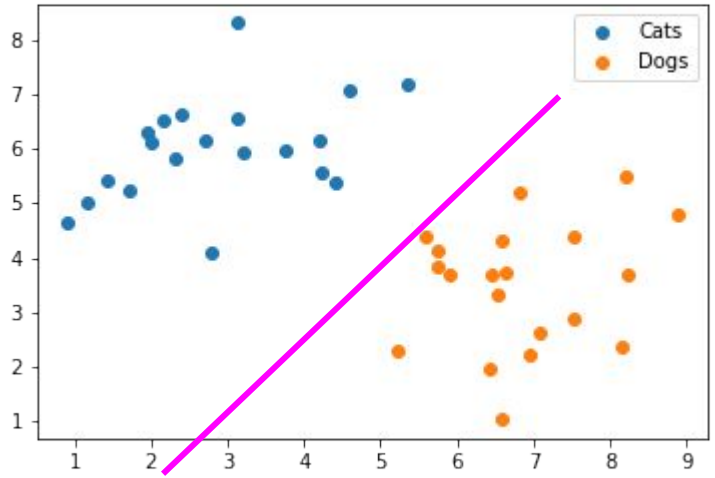
$$\hat{y}(x) = P(Y = 1|X)$$

- This is read as “Probability that the label **Y** is **1** given the features **X** we have”

So what do we need to find?

$$\hat{y}(x) = \frac{1}{1 + e^{-(W^T X + b)}}$$

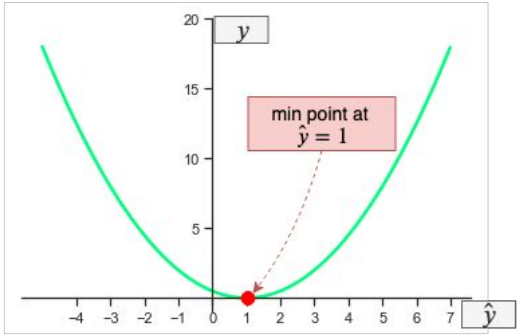
- We need to find the **decision boundary**
- That is, we must learn **W** and **b** such that an input **X** when transformed, is correctly classified as **0** or **1**
- How do we do this?
    - Gradient descent on cost function!



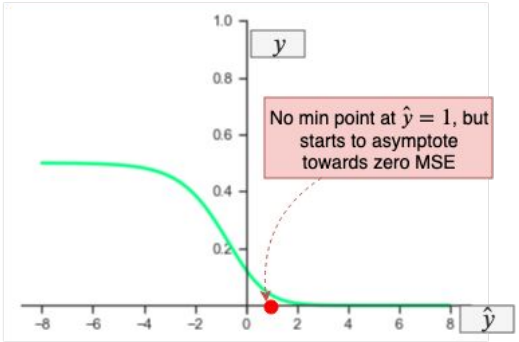
Cost Function: Why not mean squared error?

$$L(\hat{y}_1, \hat{y}_2, \ldots \hat{y}_m) = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

MSE function when $y = 1$ and $\hat{y} = (-\infty, \infty)$



MSE function when $y = 1$ and $\hat{y} = \sigma = (0, 1)$



https://towardsdatascience.com/why-using-mean-squared-error-mse-cost-function-for-binary-classification-is-a-bad-idea-933089e90df7

Cost Function: Binary Cross-Entropy Loss

- Instead, we use **Binary Cross-Entropy Loss** or **Log Loss**

$$L(\hat{y}, y) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})$$

Cost Function: Binary Cross-Entropy Loss (aka Log Loss)

$$l(\hat{y}, y) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})$$

- $\hat{y}$: prediction.
- y : label
- What happens when **y** is **1**? $l(\hat{y}, 1) = -log(\hat{y})$
- What happens when **y** is **0**? $l(\hat{y}, 0) = -\log(1 - \hat{y})$

Cost Function: Binary Cross-Entropy Loss

So the total cost across all the samples becomes:

$$L(w,b) = \frac{1}{m}\sum_{i=1}^{m} l(\hat{y}_i, y_i)$$

$$L(w,b) = \frac{1}{m}\sum_{i=1}^{m} -y_i \log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)$$

Quick poll: Correct Loss Function

Which of the following is the correct loss function for binary classification?

a. $(1 - y)\log(\hat{y}) + (y)\log(1 - \hat{y})$
b. $-(y)\log(\hat{y}) - (1 - y)\log(1 - \hat{y})$
c. $(y)\log(\hat{y}) + (1 - y)\log(1 - \hat{y})$
d. $-(1 - y)\log(\hat{y}) - (y)\log(1 - \hat{y})$

Gradient Descent

The derivatives **dL /dw** and **dL / db** are similar to those in linear regression.

$$\frac{\partial L}{\partial w} = \frac{1}{m} X^T (\hat{Y} - Y) \qquad w = w - \alpha \frac{\partial L}{\partial w}$$

$$\frac{\partial L}{\partial b} = \frac{1}{m} \sum_{i=1}^{m} (\hat{Y}_i - Y_i) \qquad b = b - \alpha \frac{\partial L}{\partial b}$$

Yhat is a column vector (m x 1) containing all the predictions, ***Y*** is a column vector (m x 1) with the labels/target, and **X** is the data (m x n)

Check out the full **derivation** if you are interested in the Math
Credit to **towardsdatascience.com**

Quick Poll: Logistic Regression Review

Which task is logistic regression well suited for?

a. Predicting the price of a house
b. Predicting whether to approve a loan or deny a loan
c. Generating pictures of dogs and cats
d. Facial recognition software

A Motivating Example

Implementing a Logistic Regression Model with SKLearn

Logistic Regression Coding Exercise

- Code along at

  https://tinyurl.com/btrack-f21-w5-demo

Thank you! We'll see you next week!

Please fill out our feedback form: bit.ly/btrack-w22-feedback

Next week: Numpy and Pandas

Very very very useful Python libraries for ML!

FB group: facebook.com/groups/uclaacmai

acm acm.ai